TECHNISCHE UNIVERSITÄT DRESDEN

Center for Information Services and High Performance Computing

Visualization of Lustre RPC Traces with Vampir

EOFS Workshop September 2011 – Paris

Zellescher Weg 12

WIL A 208

Tel. +49 351 - 463 – 34217

Michael Kluge (michael.kluge@tu-dresden.de)

ZIH

Zentrum für Informationsdienste und Hochleistungsrechnen

Content

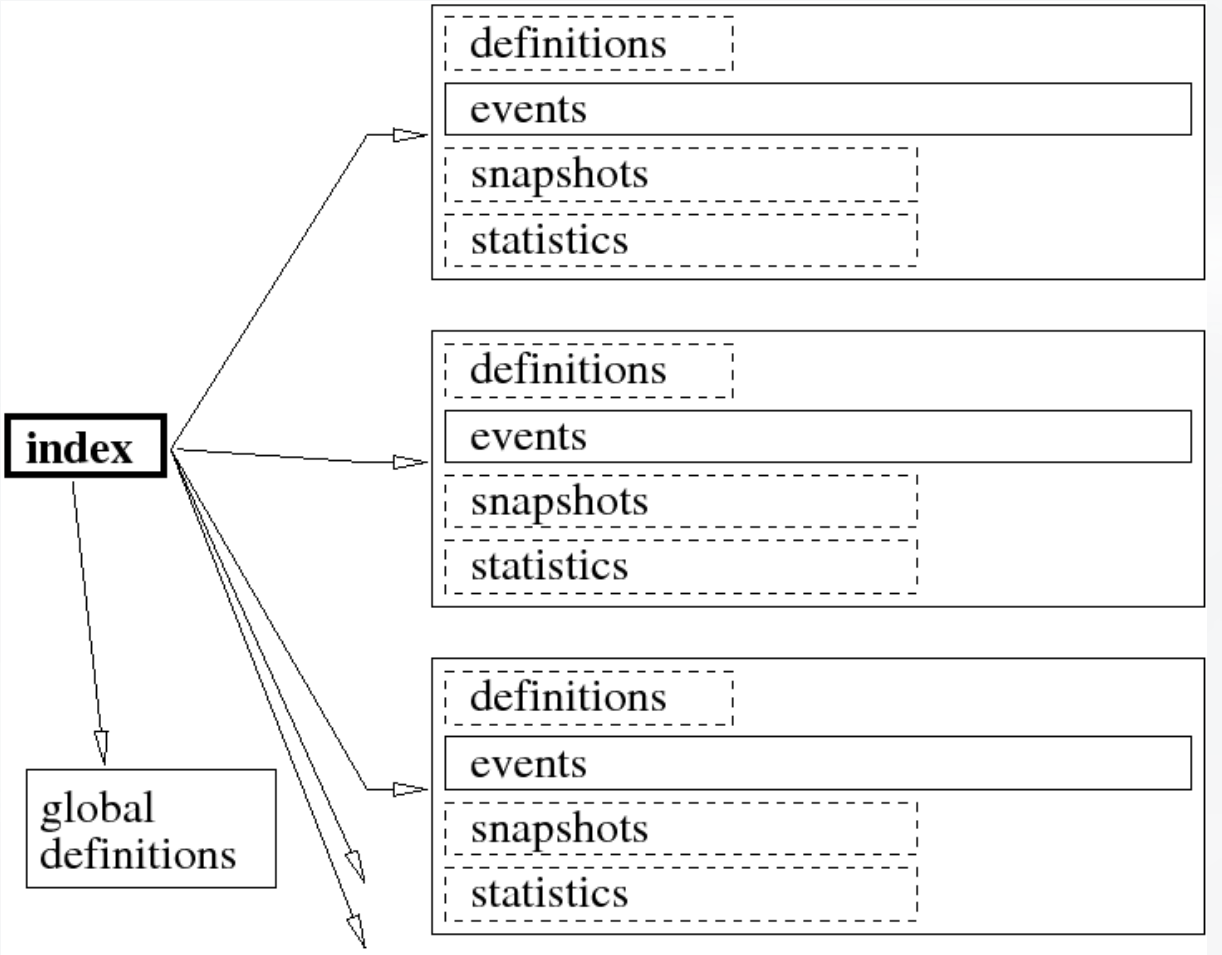
- Problem statement
- OTF and Vampir introduction
- Converter design
- Time synchronisation
- Screenshots
- Directions

Problem statement

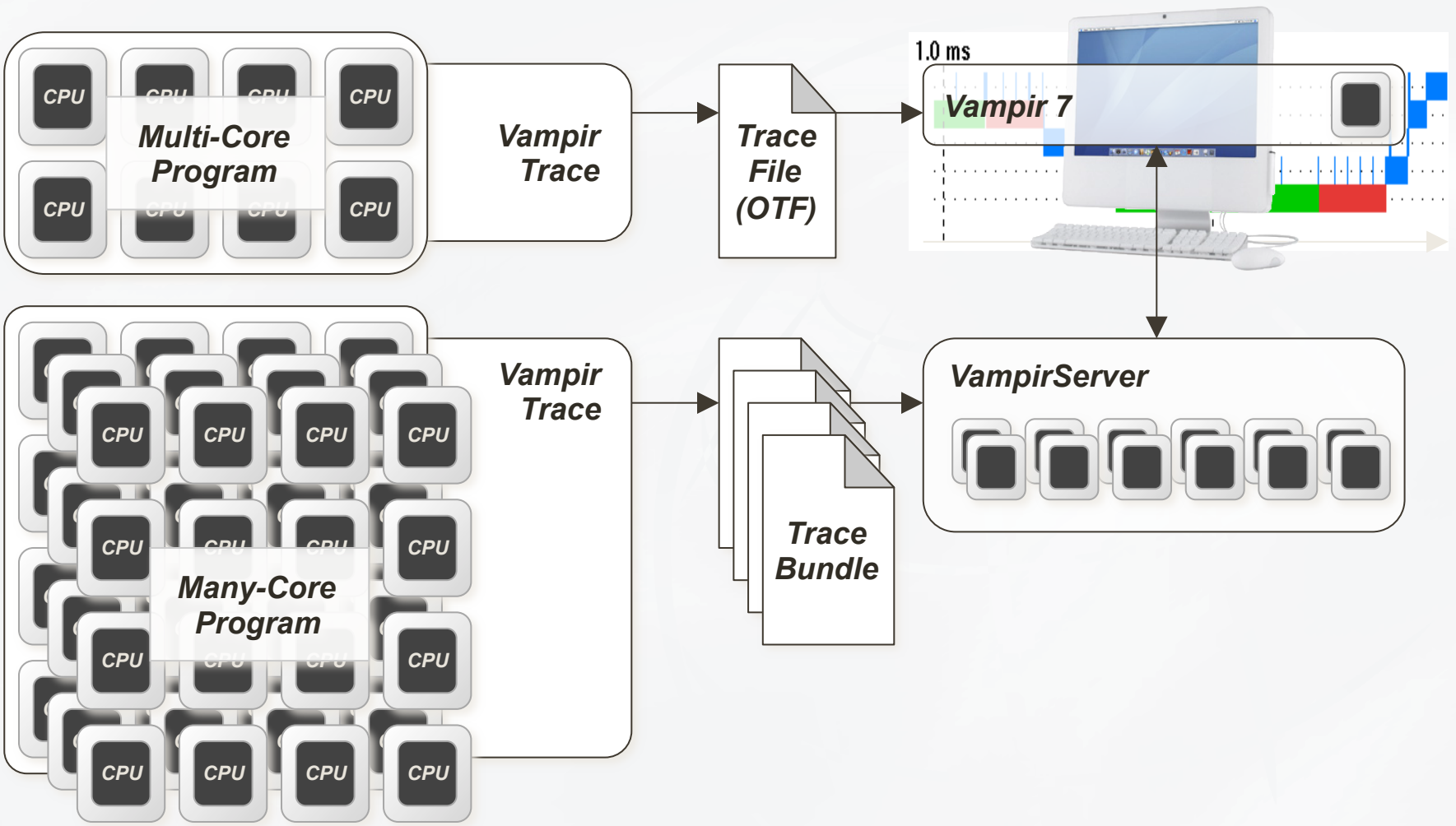
- Lustre RPC traces consist of millions of events
- Data size is in the GB range
- One trace per server and client
- Log files are text files
- Evaluation?
- Visualization?

OTF

- One index file
- One definition file
- Multiple event streams
- Each stream can store events for many processes
- Typical usage: application traces

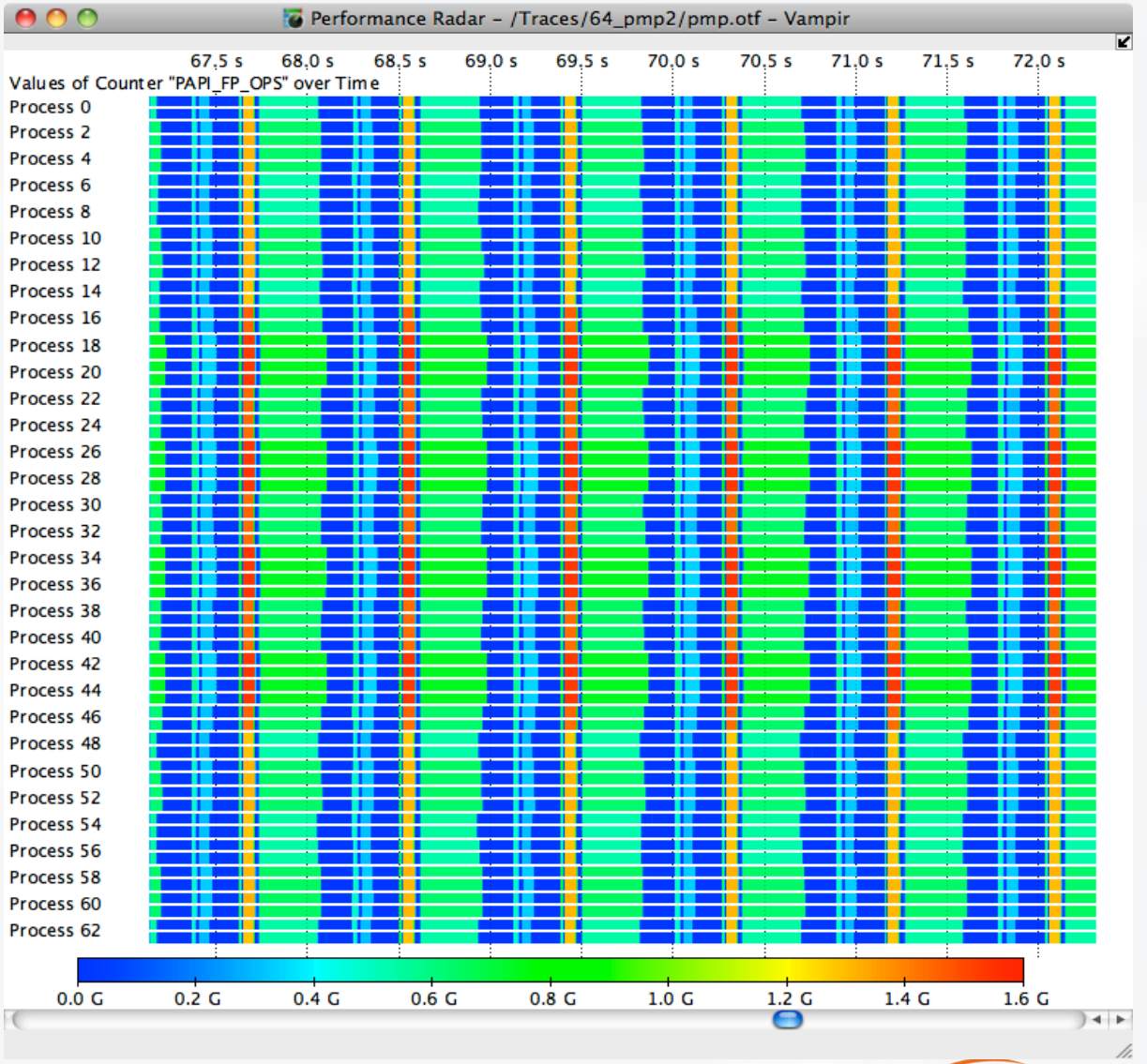


Vampir - Overview



Vampir - Performance Radar

- Color coded
- many counter timelines
- Like heat maps in gnuplot
- Basic arithmetics on counter data

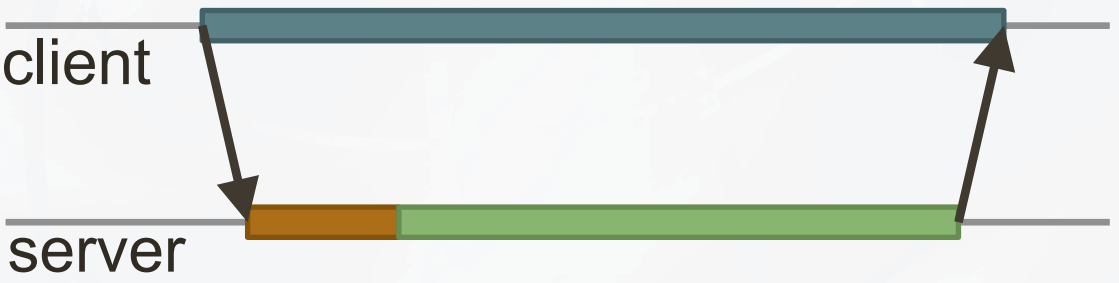


Approach

- Rewrite Lustre RPC logs to OTF trace file bundle
- Use the Performance Radar to show:
  - RPCs in flight per client
  - Average RPC completion time
  - Queue times on the server
  - Different types of RPCs
  - …

Challenges

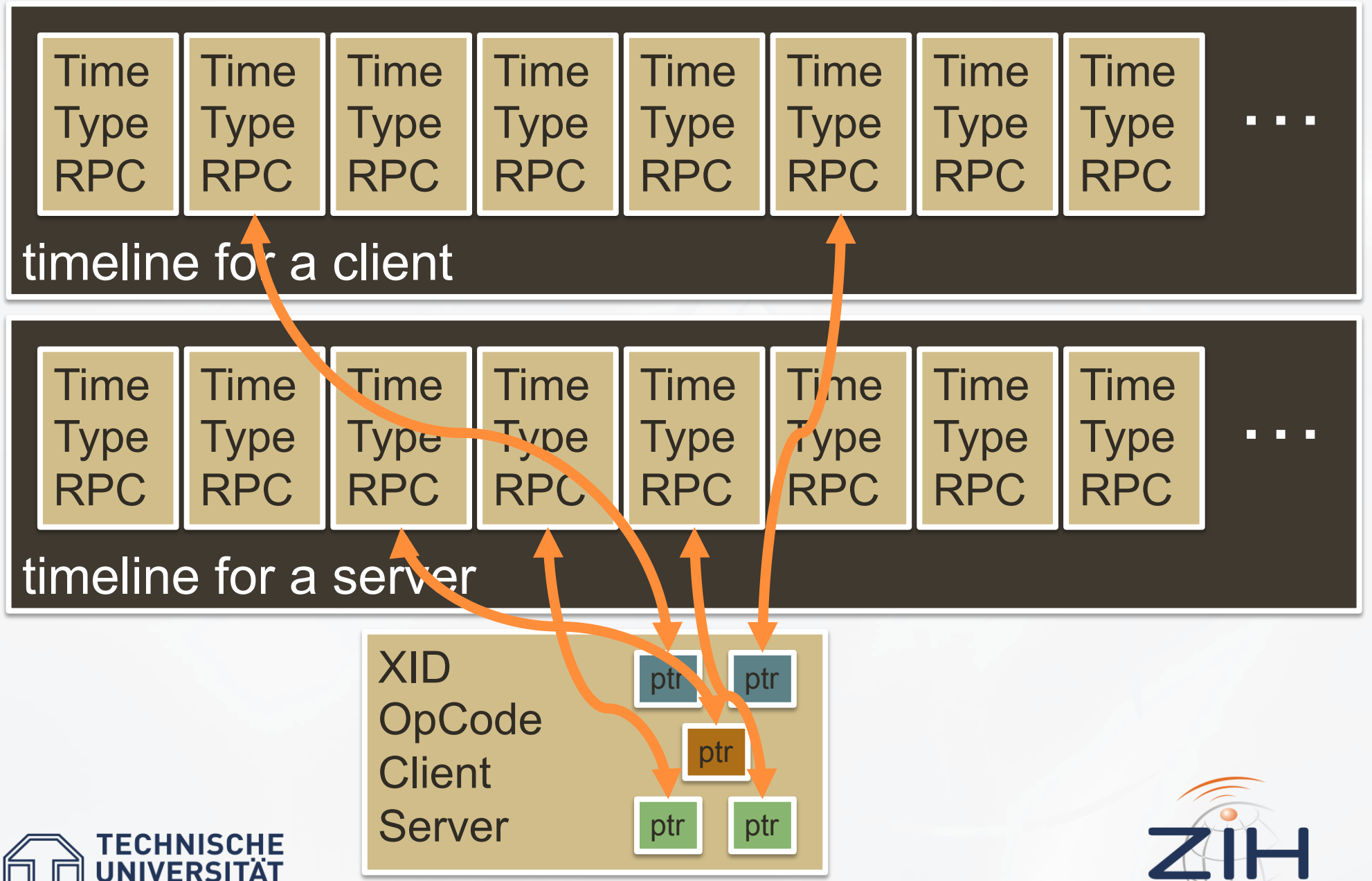
- Conversation of lots of unsynchronized individual files
- Find a ‘file name’ to ‘host name’ mapping
- Final event streams need to be ordered in time
- Five events per RPC
  - Client send
  - Server receive
  - Server start work
  - Server end work
  - Client ack
- Catch different protocol versions
- Deal with server side logs only



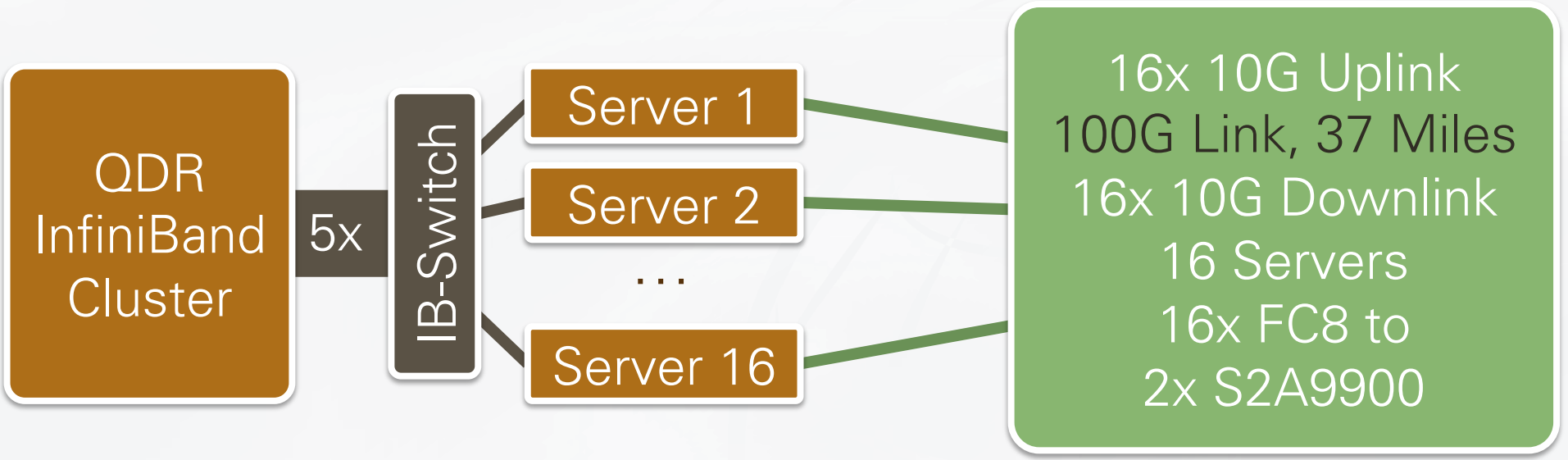
Time Synchronisation

- Initial time offsets unknown
- Unknown whether tracing started synchronously
- Solution:
  - Provide a {RPC log file, file type, NID list} tuple
  - Read the first N seconds from each file
  - Try to find matching RPC pairs
  - Create a timer offset map

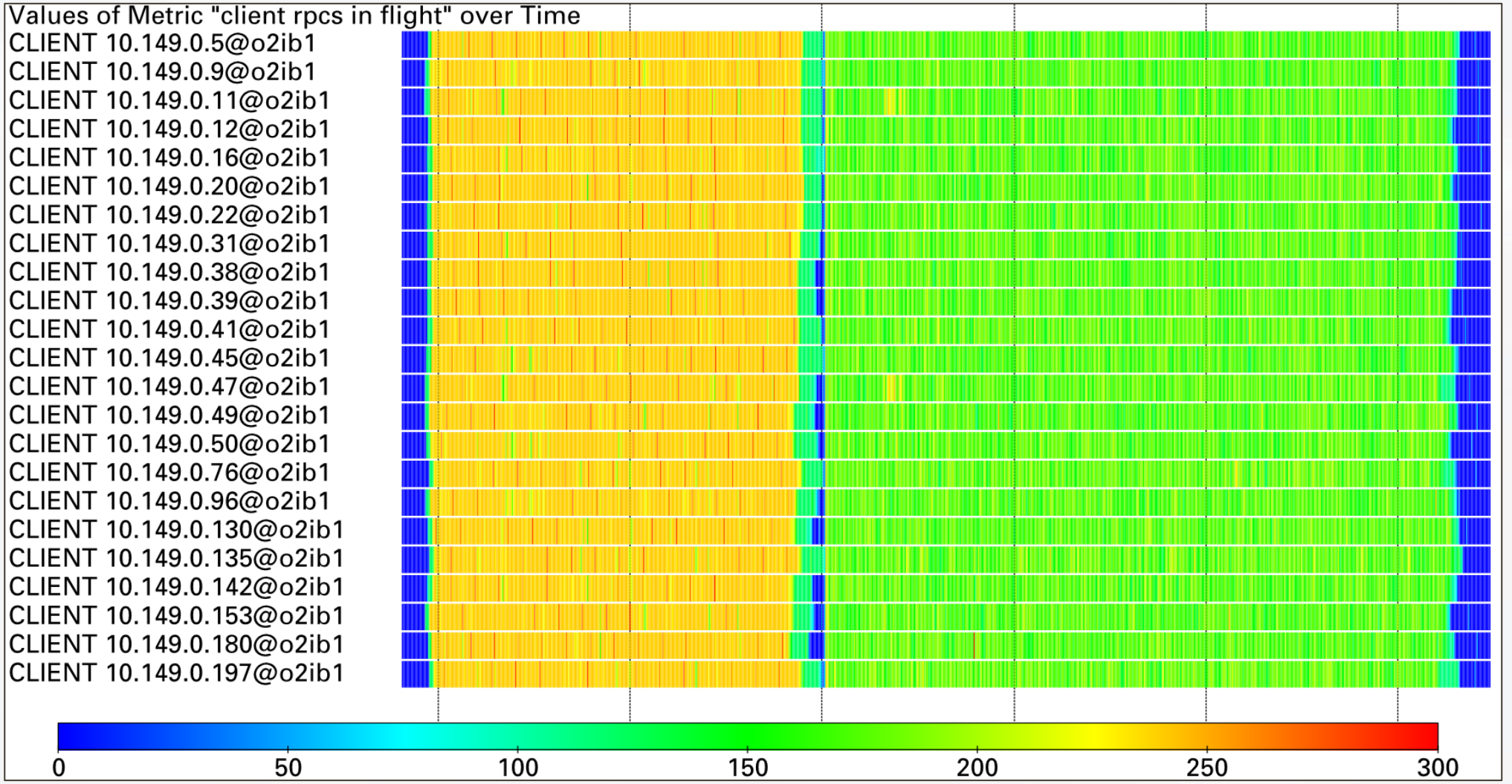
Internal Data Structure



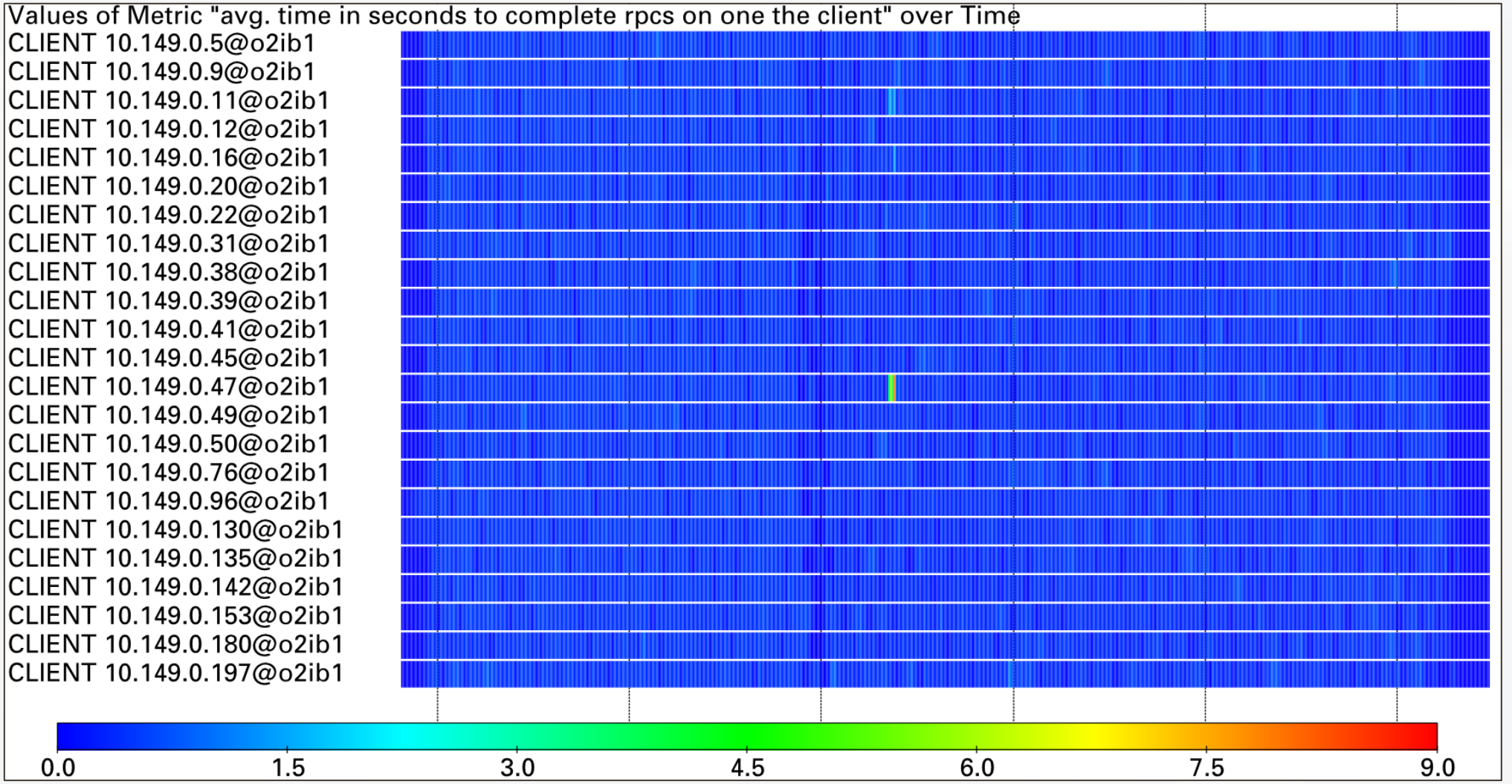
100GBit Testbed



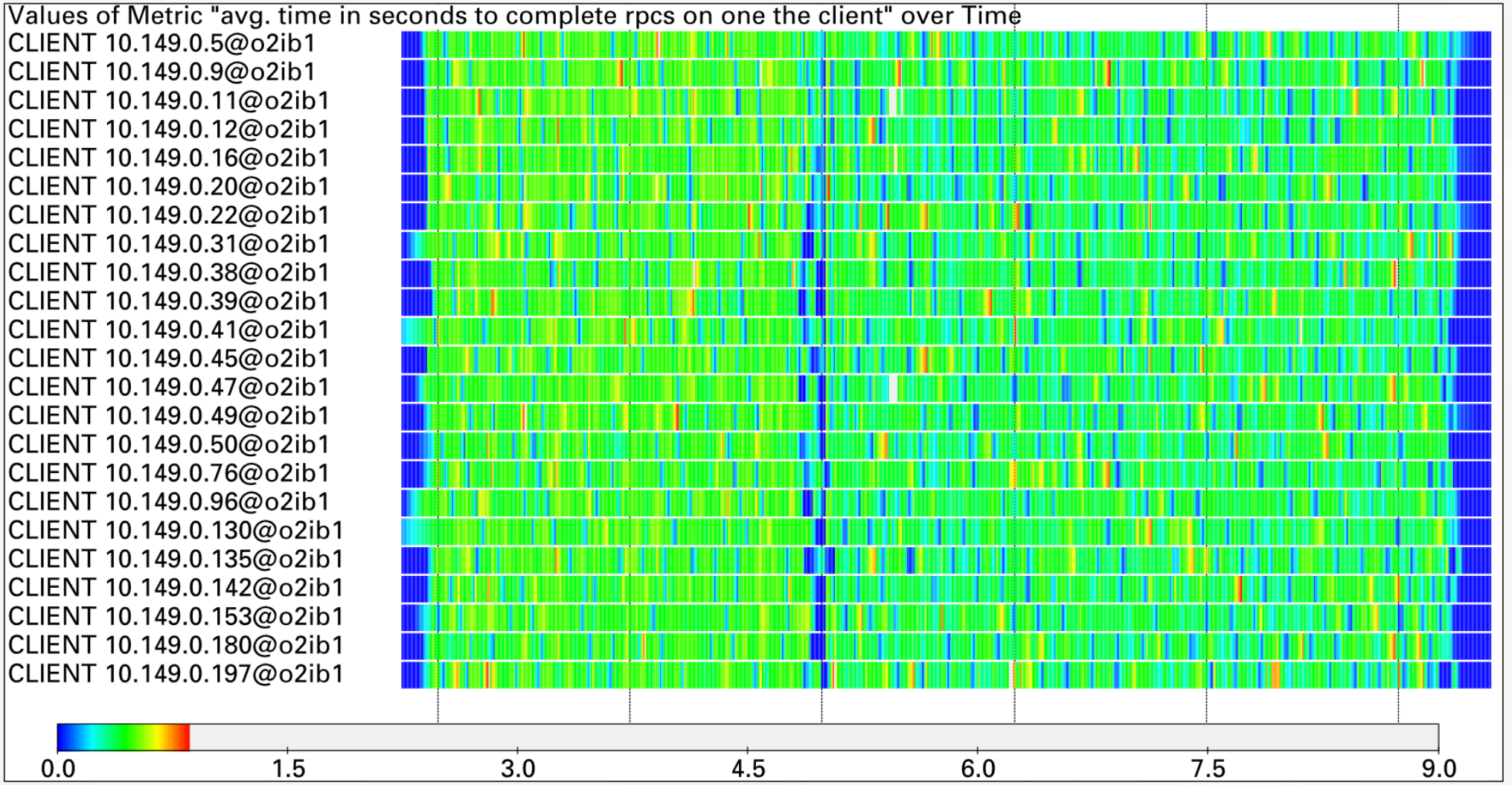
Screenshot – RPCs in flight on the clients



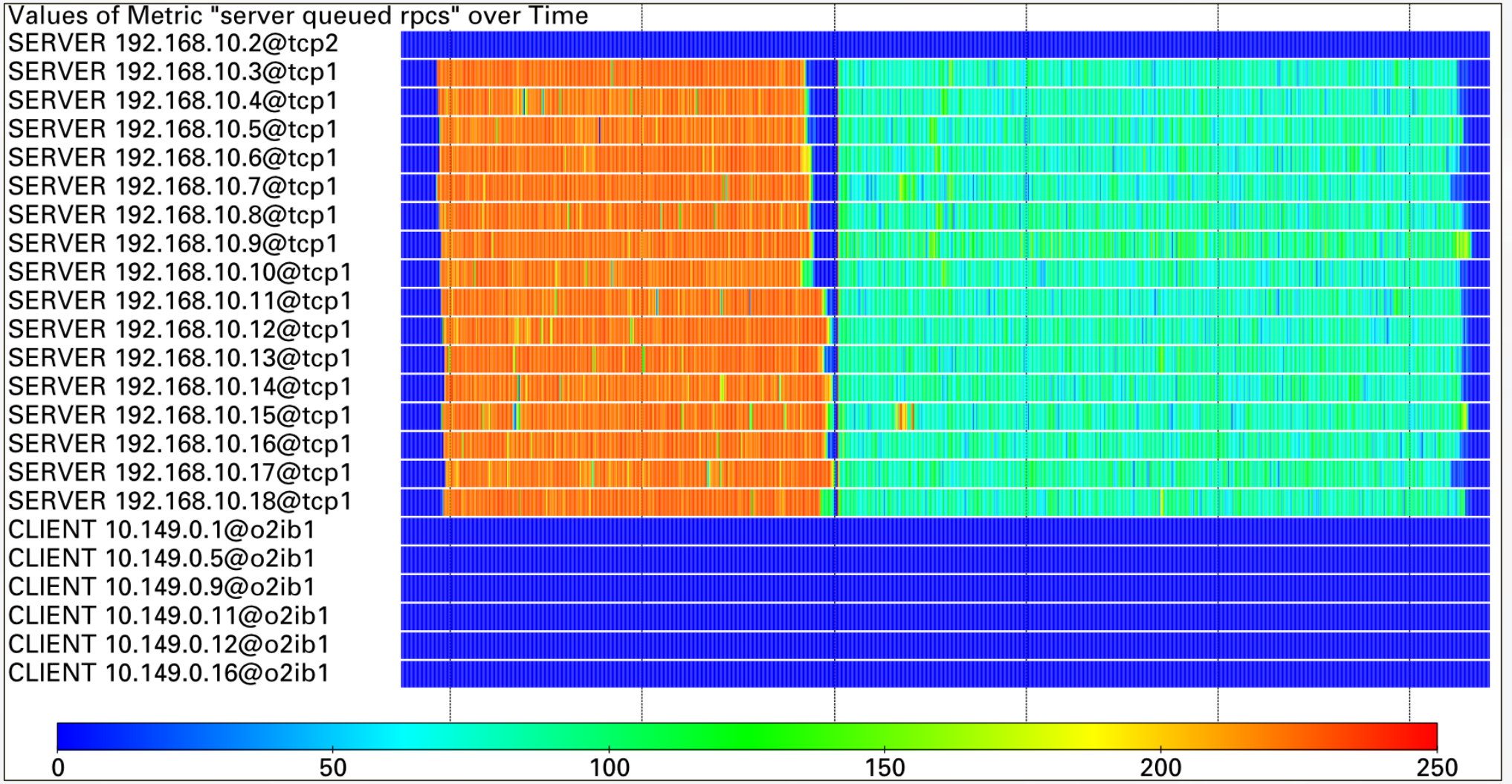
Screenshot – average time to complete an RPC



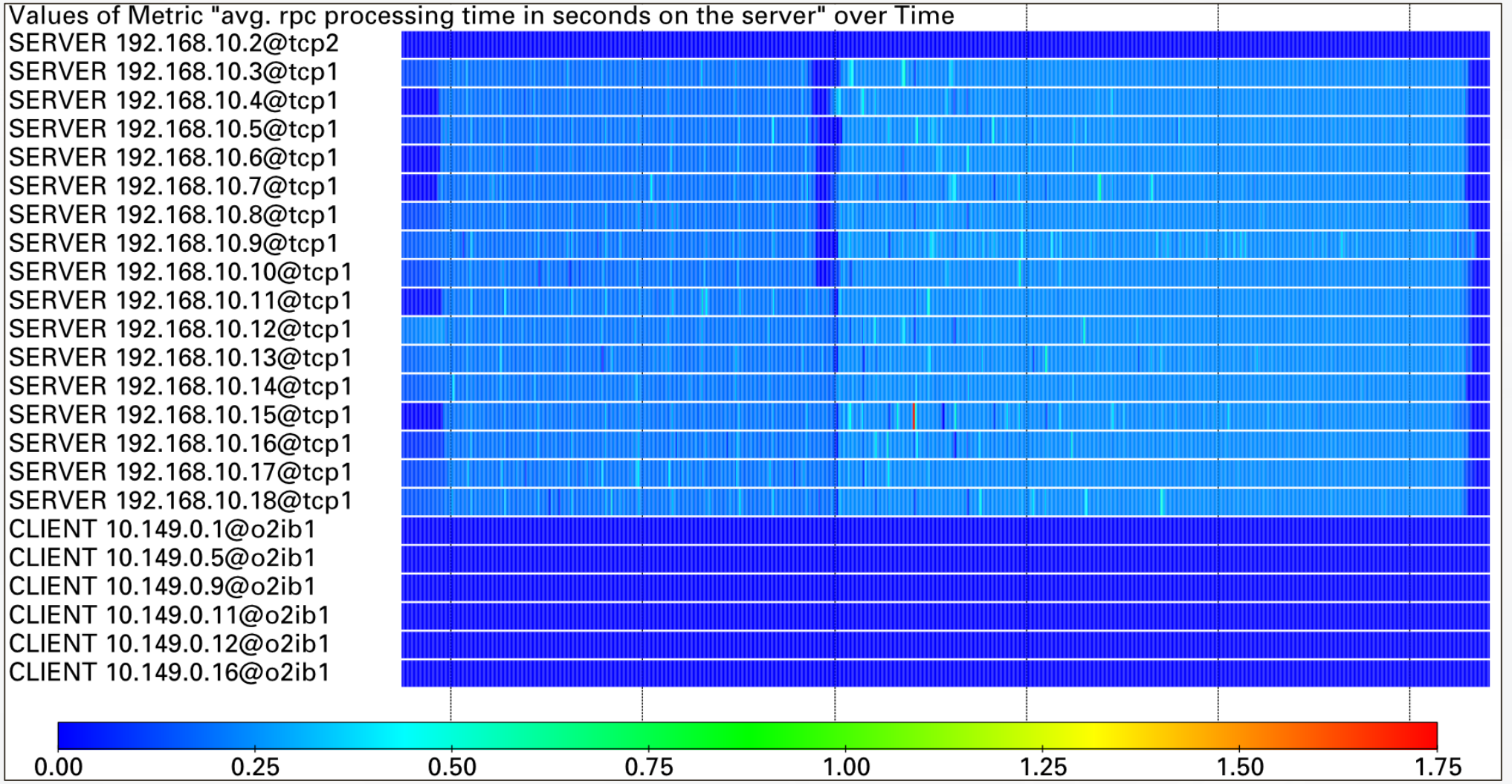
Screenshot – average time to complete an RPC



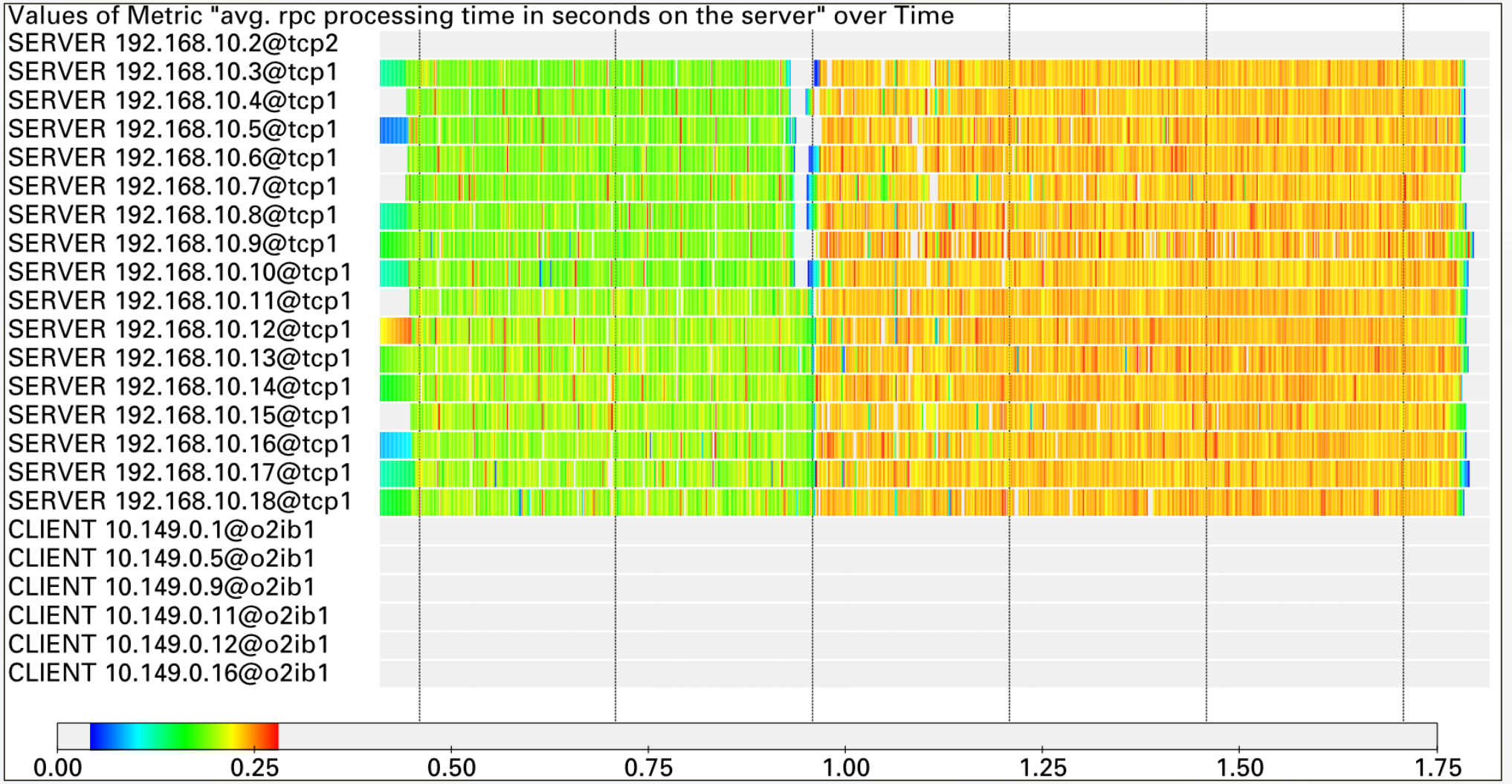
Screenshot – RPCs queued on the server



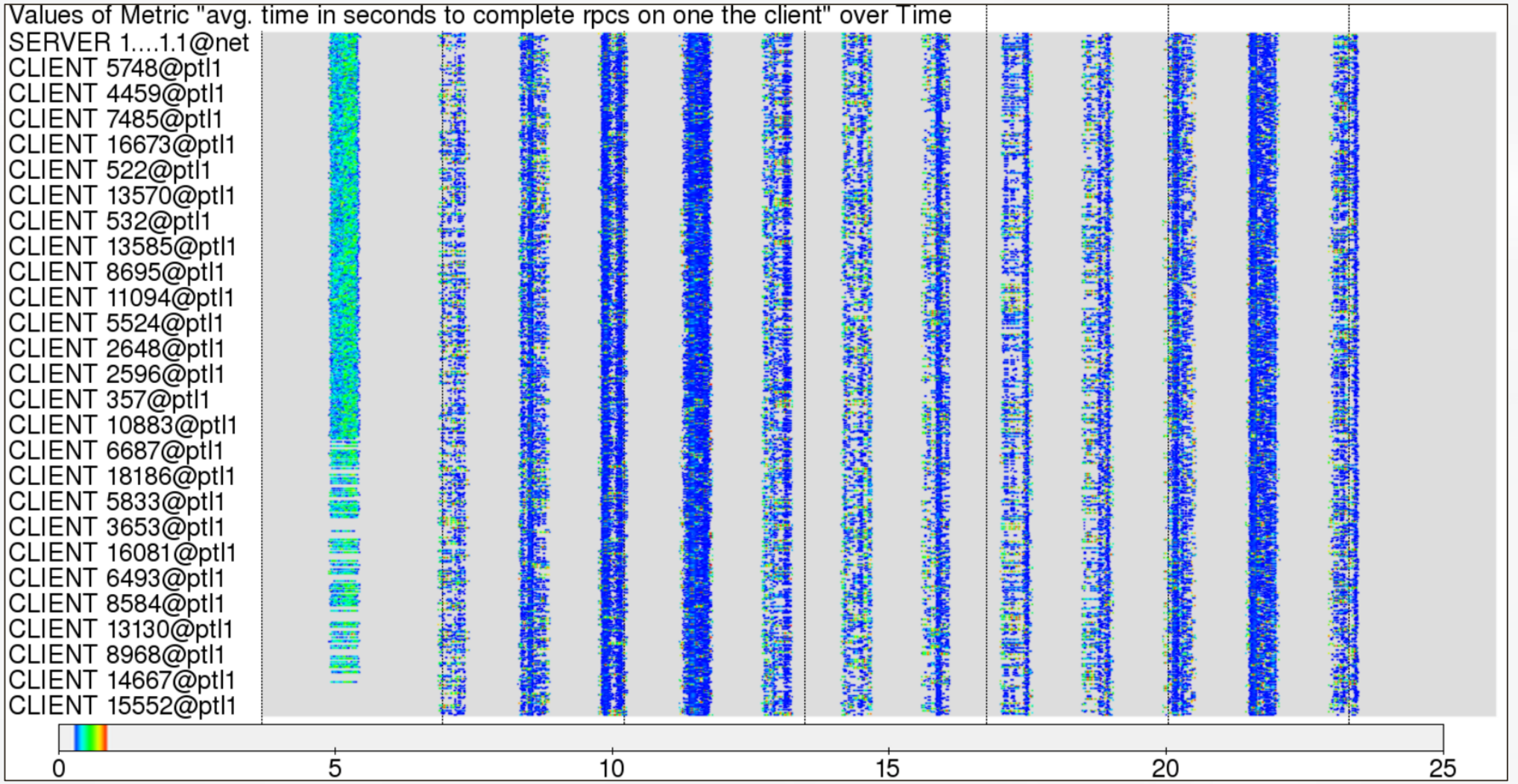
Screenshot – average RPC processing time (1)



Screenshot – average RPC processing time (2)



Screenshot – Jaguar (server log only)



Where to go from here

- More metrics needed?
- Use markers to mark interesting locations
- Locks are in the log files
- How about a complete log from jaguar
- Other possibilities to explore:
  - make use of the Python interface for OTF
  - make use of the Octave import for OTF files

Time for Questions

Some internal limits

- Queued (open) RPCs per process
- Maximum RPC completion time
- Time window for average calculations
- Maximum allowed time synchronisation